بسمه تعالی

تمرین سری نهم درس یادگیری عمیق

پوریا محمدی نسب

۴۰۰۷۲۲۱۳۸

فهرست

سوال ۱. ..... ۳

سوال ۲. ..... ٦

الف. ٦

ب. ۷

سوال ۳. ..... ۸

الف. ۸

ب. ۹

References ۱۱

سوال ۱.

در این مقاله، از یک شبکه عصبی بازگشتی برای آموزش جانمایی عبارات استفاده شده و سپس عملکرد آن در ترجمه ماشینی سنجیده شده است. معماری استفاده شده در این مقاله را توضیح دهید.(برای پاسخ دادن به این سوال مطالعه قسمتهای ابتدایی مقاله و بررسی اجمالی معماری آن کافی است اما مطالعه کامل این مقاله کاربردی توصیه میشود.)

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).

معماری این شبکه عصبی بازگشتی به صورت رمزنگار (encoder) و رمزگشا (decoder) است که قسمت encoder آن وظیفه دارد تا دنباله ای از نمادها(symbols) را به یک آرایه (vector) با طول ثابت تبدیل کند. قسمت دوم این معماری که decoder آن را شامل میشود وظیفه دارد تا بازنمایی خارج شده از encoder را به دنباله ی جدیدی از نمادها تبدیل کند. حال به ذکر جزئیات بیشتری از مقاله میپردازیم. از نکات کلیدی آموزش این مدل میتوان به این موضوع اشاره کرد که هر دو قسمت شبکه باید به طور همزمان آموزش ببینند. علت این همزمانی آموزش بیشینه کردن احتمال شرطی دنباله نمادی است که در decoder باید تولید شود که این احتمال شرطی به دنباله ورودی بشدت وابسته است. از نوآوری های این شبکه میتوان به واحد مخفی معرفی شده اشاره کرد که این واحد مخفی میتواند به صورت تطبیقی باعث فراموشی یا به یاد ماندن شود. شبکه ارائه شده در مقاله از دید آماری، مدلی کلی (general) است که توزیع شرطی دنباله هایی با طول مختلف را از روی توزیع شرطی دنباله های دیگر یاد میگیرد که به صورت ریاضی به شکل زیر بیان میشود:

$$p(y_1, \ldots, y_{T'} \mid x_1, \ldots, x_T)$$

باید به این نکته توجه کرد که T و T′ لزوما برابر نیستند به طور شهودی نیز قابل درک است که طول یک جمله از یک زبان در هنگام ترجمه به زبان دیگر ممکن است طول متفاوتی داشته باشد. برای درک بهتر چگونگی عملکرد encoder باید توجه داشت که هر نماد در دنباله ورودی X وارد رابطه زیر میشود و تغییر میکند تا در حالت مخفی (hidden state) شبکه بازگشتی ذخیره شود و این کار را تا انتهای جمله که به کاراکتر پایان جمله (end-of-sequence) میرسد انجام میدهد و در این مرحله حالت مخفی شبکه در واقع یک خلاصه از کل دنباله ورودی در نظر گرفته میشود که با c نمایش میدهیم.

$$\mathbf{h}_{\langle t \rangle} = f\left(\mathbf{h}_{\langle t-1 \rangle}, x_t\right)$$

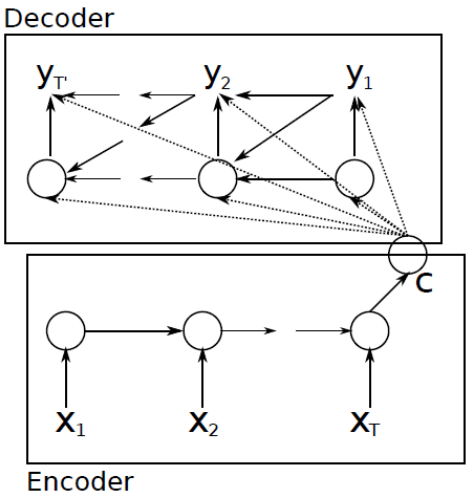
قسمت decoder مدل پیشنهادی نیز یک شبکه RNN که برای تولید دنباله خروجی با پیشبینی نماد بعدی با توجه به حالت مخفی آموزش دیده است. البته این قسمت از شبکه با یک RNN معمولی کمی تفاوت دارد زیرا حالت مخفی (h_t) و خروجی احتمالی (y_t) بر اساس احتمال خروجی احتمالی قبلی (y_{t-1}) و خلاصه دنباله ورودی (c) تولید میشوند. پس حالت مخفی این decoder جدید در زمان t از رابطه زیر محاسبه میشود:

$$\mathbf{h}_{\langle t \rangle} = f\left(\mathbf{h}_{\langle t-1 \rangle}, y_{t-1}, \mathbf{c}\right)$$

و به تبع آن توزیع شرطی نماد بعدی نیز به صورت زیر برابر است با:

$$P(y_t|y_{t-1}, y_{t-2}, \ldots, y_1, \mathbf{c}) = g\left(\mathbf{h}_{\langle t \rangle}, y_{t-1}, \mathbf{c}\right)$$

شکل زیر نمایی کلی از شبکه پیشنهادی مقاله است:



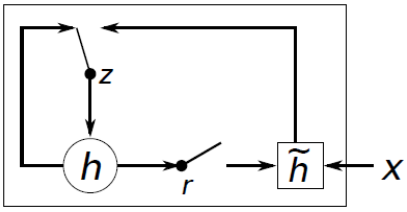
همانطور که در ابتدای توضیحات نیز اشاره شد در این مدل پیشنهادی هر دو قسمت شبکه باید با هم آموزش ببینند و objective function این آموزش این است که log-likelihood شرطی بیشینه شود.

$$\max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \log p_{\boldsymbol{\theta}}(\mathbf{y}_n \mid \mathbf{x}_n)$$

علاوه بر معماری جدید این مقاله، یک hidden unit جدید نیز معرفی شد. در شبکه های بازگشتی معمولی برای محاسبه hidden state از تابع زیر استفاده میشود:

$$\mathbf{h}_{\langle t \rangle} = f\left(\mathbf{h}_{\langle t-1 \rangle}, x_t\right)$$

که در آن f یک تابع فعال سازی غیر خطی است که میتواند یک تابع ساده یا یک LSTM پیچیده باشد. که LSTM میتواند بسیار موثر باشد اما در عین حال محاسبات سنگینی دارد. در این مقاله یک f جدید پیشنهاد میشود که از لحاظ کارکرد به LSTM بسیار شبیه است اما در عین حال برای محاسبه و پیاده سازی بسیار ساده تر است که شکل و رابطه آن را در زیر میبینید:



$$r_j = \sigma\left(\left[\mathbf{W}_r \mathbf{x}\right]_j + \left[\mathbf{U}_r \mathbf{h}_{\langle t-1 \rangle}\right]_j\right)$$

که در رابطه بالا σ تابع logistic sigmoid است. و اندیس j به معنای j امین المان یک آرایه است. به صورت مشابه برای قسمت Z در شکل داریم:

$$z_j = \sigma\left([\mathbf{W}_z\mathbf{x}]_j + [\mathbf{U}_z\mathbf{h}_{\langle t-1\rangle}]_j\right)$$

در انتها برای محاسبه hidden state داریم :

$$h_j^{\langle t\rangle} = z_j h_j^{\langle t-1\rangle} + (1 - z_j)\tilde{h}_j^{\langle t\rangle},$$

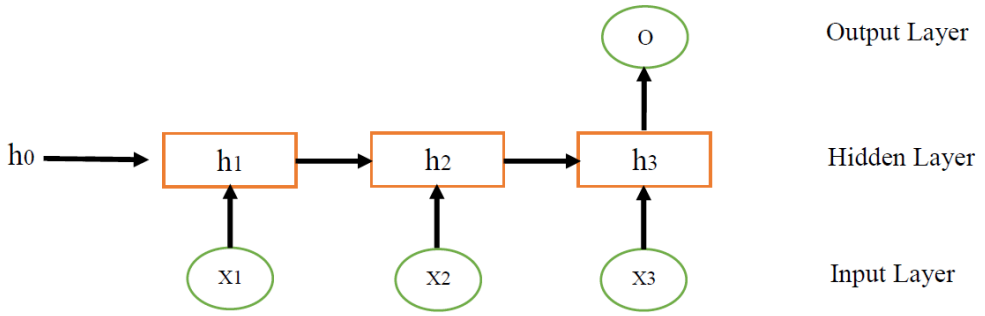
در انتها نتایج روش پیشنهادی را در مقایسه با شبکه RNN معمولی و چند روش دیگر مشاهده میکنیم. در جدول زیر WP به معنی جریمه ی کلمه ای (Word Penalty) است که به معنی تعداد کلمات ناشناخته برای شبکه است.

Models	BLEU	
	dev	test
Baseline	30.64	33.30
RNN	31.20	33.87
CSLM + RNN	31.48	34.64
CSLM + RNN + WP	31.50	34.54

سوال ۲.

الف.

هدف از این تمرین آشنایی با Backpropagation در شبکه های بازگشتی است. معماری داده شده یک شبکه ساده بازگشتی را نشان میدهد که لایه های ورودی، پنهان و خروجی در آن مشخص شده است. تابع ضرر را MSE در نظر بگیرید. تمام مراحل به روزرسانی را برای دو وزن Wxh و Whh به صورت دستی بنویسید.(وزن Wxh و وزن Whh به ترتیب در لایه ورودی و لایه پنهان مشترک هستند. از تابع فعالسازی صرف نظر کنید).



در ابتدا شبکه صورت سوال را بررسی میکنیم. در این شبکه h1، h2 و h3 حالت های مخفی(hidden states) شبکه به ترتیب در لحظه t1، t2 و t3 هستند. وزن بین h0 و h1 که با W_{hh} نمایش میدهیم نیز ماتری وزن است که طبق گفته صورت سوال در لایه پنهان مشترک است. x1، x2 و x3 ورودی های شبکه هستند و ماتریس وزن آنها که مشترک نیز هست را با W_{xh} نمایش میدهیم. در انتها وزن بین خروجی O و حالت مخفی h3 را با W_y نمایش میدهیم.

برای هر زمان $t > 2$ روابط زیر برقرار است:

$$S_t = g1(W_x x_t + W_s S_{t-1})$$

$$Y_t = g2(W_Y S_t)$$

در رابطه بالا g1 و g2 توابع فعال سازی هستند که البته در صورت سوال گفته شده است از آنها صرف نظر کنیم. حال میخواهیم backpropagation را در t=3 انجام دهیم. تابع خطا MSE است پس داریم:

$$E_t = (d_t - Y_t)^2$$

که در مقادله بالا d_t مقدار خروجی مورد نظر است. برای انجام عمل BP باید وزن ورودی ها، حالت های مخفی و خروجی را تنظیم کنیم. در ابتدا به تنظیم وزن W_y میپردازیم. طبق رابطه زنجیری داریم:

$$\frac{\partial E_3}{\partial W_y} = \frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial W_y}$$

در ادامه برای تنظیم وزن hidden state ها باید قاعده زنجیری را برای هر s محاسبه و با هم جمع کنیم:

$$\frac{\partial E_3}{\partial W_s} = \left(\frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial S_3} \times \frac{\partial S_3}{\partial W_s}\right) + \left(\frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial S_3} \times \frac{\partial S_3}{\partial S_2} \times \frac{\partial S_2}{\partial W_s}\right) + \left(\frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial S_3} \times \frac{\partial S_3}{\partial S_2} \times \frac{\partial S_2}{\partial S_1} \times \frac{\partial S_1}{\partial W_s}\right)$$

که در حالت کلی اگر N حالت مخفی داشتیم فرمول آن به صورت زیر بود:

$$\frac{\partial E_N}{\partial W_s} = \sum_{i=1}^{N} \frac{\partial E_N}{\partial Y_N} \times \frac{\partial Y_N}{\partial S_i} \times \frac{\partial S_i}{\partial W_s}$$

حال برای تنظیم وزن ورودی ها داریم:

$$\frac{\partial E_3}{\partial W_X} = \left(\frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial S_3} \times \frac{\partial S_3}{\partial W_X}\right) + \left(\frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial S_3} \times \frac{\partial S_3}{\partial S_2} \times \frac{\partial S_2}{\partial W_X}\right) + \left(\frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial S_3} \times \frac{\partial S_3}{\partial S_2} \times \frac{\partial S_2}{\partial S_1} \times \frac{\partial S_1}{\partial W_X}\right)$$

که برای این حالت نیز در حالت کلی داریم:

$$\frac{\partial E_N}{\partial W_X} = \sum_{i=1}^{N} \frac{\partial E_N}{\partial Y_N} \times \frac{\partial Y_N}{\partial S_i} \times \frac{\partial S_i}{\partial W_X}$$

ب.

مشکل Vanishing gradient در شبکه های بازگشتی را توضیح دهید و روش های حل آن را نام ببرید.

در شبکه های RNN هنگامی که مقدار وزن ها بشدت و به صورت نمایی کاهش میابد و به صفر نزدیک میشود مشکلی که به وجود می آید این است که شبکه قدرت یادگیری بعضی از وابستگی ها و حالات مخفی در فاصله زمانی زیاد را از دست خواهد داد.

یک راه حل حرکت دادن window آموزشی در طول فرآیند آموزش است. در BP تمام دنباله در مرحله forward و backward در نظر گرفته میشوند تا loss و گرادیان محاسبه شود. اما ما با این تکنیک فقط قسمتی از دنباله را در نظر میگیریم.

روش های رایج دیگری مانند استفاده از تابع فعال سازی ReLU و استفاده از معماری LSTM به دلیل وجود گیت فراموشی میتوانند موثر باشند.

سوال ۳.

الف.

دو مجموعه آموزشی و آزمایشی مشخص شده اند. باتوجه به جملات مجموعه آموزشی، احتمال جملات آزمایشی را محاسبه کنید. لطفا تمام مراحل محاسبات خود را یادداشت نمایید(تمام محاسبات به صورت دستی انجام شود).

Training corpus
<s> I like deep learning class </s>
<s> I am a student </s>
<s> student has different lessons </s>
<s> student likes learning different languages </s>

Test data
<s> I like learning different lessons </s>
<s> student likes deep learning class </s>

*** راهنمایی:**

$$P(W_i \mid W_{i-1}) = \frac{count(W_{i-1}, W_i)}{count(W_{i-1})}$$

برای تخمین زدن مقدار احتمال این جملات از maximum likelihood estimation یا به اختصار MLE استفاده میکنیم.

برای این کار تعداد هر گزاره را شمارش میکنیم و آن ها بین ۰ و ۱ نرمالسازی میکنیم.

محاسبه احتمالات مورد نیاز برای محاسبه احتمال جملات آزمایشی:

$$P(I \mid <s>) = \frac{2}{4} = 0.5, \quad P(like \mid I) = \frac{1}{2} = 0.5, \quad P(learninig \mid like) = \frac{1}{2} = 0.5,$$

$$P(different \mid learning) = \frac{1}{2} = 0.5, \quad P(lessons \mid different) = \frac{1}{2} = 0.5,$$

$$P(</s> \mid lessons) = \frac{1}{1} = 1, \quad P(student \mid <s>) = \frac{2}{4} = 0.5,$$

$$P(likes \mid student) = \frac{1}{3} = 0.33, \quad P(deep \mid likes) = \frac{1}{2} = 0.5,$$

$$P(learning \mid deep) = \frac{1}{2} = 0.5, \quad P(class \mid learning) = \frac{1}{2} = 0.5,$$

$$P(</s> \mid class) = \frac{1}{1} = 1$$

حال برای محاسبه احتمال هر جمله آزمایشی باید احتمال دو به دوی هر دو عبارت پشت سر هم در گزاره را در هم ضرب کنیم:

برای جمله اول:

P(<s> I like learning different lessons </s>) =

P(I | <s>) * P(like | I) * P(leaning | like) * P(different | learning) * P(lessons | different) * P(</s> | lessons) =

0.5 * 0.5 * 0.5 * 0.5 * 0.5 * 1 = 0.03125

برای جمله دوم:

P(<s> student likes deep learning class </s>) =

P(student | <s>) * P(likes | student) * P(deep | likes) * P(learning | deep) * P(class | learning) * P(</s> | class) =

0.5 * 0.33 * 0.5 * 0.5 * 0.5 * 1 = 0.020833

ب.

مدلهای مارکوف با مرتبه های ۰و۱و۲ را با یکدیگر مقایسه کنید. مزایا و معایب هرکدام را توضیح دهید.

به مدل هایی که به تعداد کمی از نمونه های قبلی وابسته باشند مدل های کارکوف میگوییم.

مدل مارکوف مرتبه ۰ به unigram نیز معروف است و هر x کاملا مستقل از باقی x ها در نظر گرفته میشود:

$P(x_1, x_2, ..., x_n) = P(x_1)P(x_2)...P(x_n)$

مدل مارکوف مرتبه ۱ به bigram معروف است که هر x وابسته به تنها یک x قبل از خودش است و احتمالات شرطی در نظر گرفته میشوند.

$P(x_1, x_2, ..., x_n) = P(x_1)P(x_2|x_1)...P(x_n|x_{n-1})$

مدل مارکوف مرتبه ۲ به trigram معروف است که هر x ورودی به حد اکثر ۲ ورودی x قبل از خود وابسته است و در این حالت نیز مانند bigram احتمالات از جنس احتمالات شرطی هستند.

$P(x_1, x_2, ..., x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1 , x_2)...P(x_n|x_{n-2} , x_{n-1})$

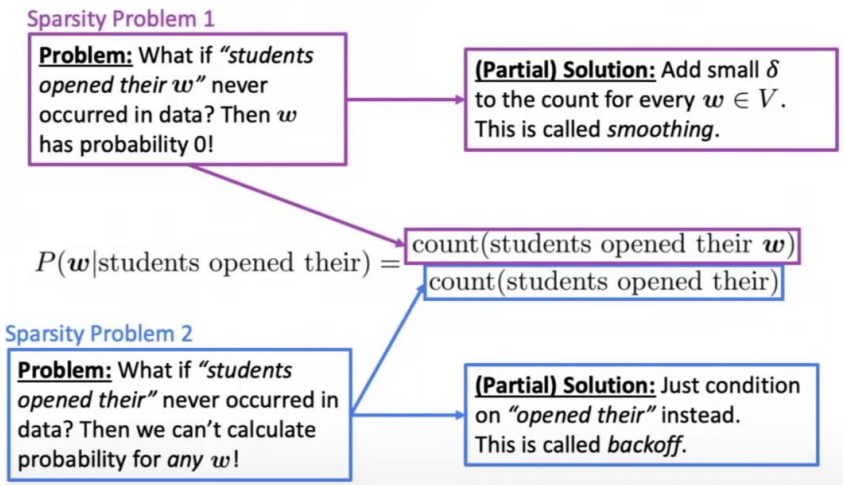
برای مقایسه خوبی و بدی هر مرتبه میتوان گفت اگر n که مرتبه مدل مارکوف است کم باشد کلمات در یک گزاره مستقل تر از هم دیگر در نظر گرفته میشوند و مشکلاتی از جمله بی معنی بودن عبارت های زبان را به همراه دارد. هر چه n عدد بزرگتری شود پراکندگی (sparsity) افزایش میابد و با ۲ مشکل اساسی مواجه میشویم. مشکل اول این است که در یک n-gram کل عبارت یافت نشود بدین معنی که کلمه x_n هیچگاه پشت کلمات x_{n-1} و x_{n-2} و ... ظاهر نشود که در این حالت احتمال کل جمله صفر میشود.(راه حل : اضافه کردن یک مقدار کوچک ε به احتمال همه ی کلمات که به روش smoothing معروف است.) برای مثال فرض کنید قصد داریم در یک مدل با مرتبه ۳ (trigram) احتمال شرطی جمله زیر را محاسبه کنیم:

My research interest is deep learning.

اما ممکن است عبارت "interest is deep" هیچگاه اتفاق نیوفتاده باشد پس در ضرب احتمالات شرطی یک جمله صفر میشود.

مشکل دوم که با مرتبه بالا اتفاق میوفتد ممکن است کلماتی که باید به احتمال شرطی آنها با هم محاسبه شوند تا احتمال کلمه x بعد از آن را در نظر بگیریم هیچگاه اتفاق نیوفتاده باشد که در این صورت نمیتوانیم مقدار احتمال شرطی را محاسبه کنیم. (راه حل: بر روی قسمت کوچک تری از عبارت احتمال شرطی را محاسبه کنیم که به این روش backoff میگویند.) برای مثال همان جمله مثال قبلی را در نظر بگیرید. در این حالت ممکن است عبارت "interest is" هیچگاه اتفاق نیوفتاده باشد پس ما نمیتوانیم P(deep | interest, is) را حساب کنیم.

شکل زیر مشکل بالابردن مرتبه n را بهتر نمایش میدهد.



References

1) https://www.geeksforgeeks.org/ml-back-propagation-through-time/

2) https://www.codingninjas.com/codestudio/library/backpropagation-through-time-rnn

3) Ribeiro, A. H., Tiels, K., Aguirre, L. A., & Schön, T. (2020, June). Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness. In *International Conference on Artificial Intelligence and Statistics* (pp. 2370-2380). PMLR.

4) https://towardsdatascience.com/the-exploding-and-vanishing-gradients-problem-in-time-series

5) https://web.stanford.edu/~jurafsky/slp3/3.pdf

6) https://medium.com/@rachel_95942/language-models-and-rnn-c516fab9545b